Lossless Speech Compression Techniques: A Literature Review

Pratibha Warkade,Agya Mishra

Abstract--- **Nowadays, computer technology mostly focusing on storage space and speed. With the rapid growing of important data and increased numbers of applications, devising new approach for efficient compression and encryption methods are playing a vital role in performance. In this literature review, burrows wheeler transformation (BWT) is introduced for pre processing of the input data and made several performance analysis experiments over different compression techniques like Lempel-Ziv 77, Lempel-Ziv-Welch, Prediction with partial match, Move to front coding, j-bit encoding along with various entropy coding .And improved compression ratio has been found by applying BWT as pre-processing step, hence it will be good for further research work.**

Keywords---**Arithmetic, BWT, Huffman coding, IDBE, Jbitencoding, LZ77, LZW, MTF, PPM, Runlength coding, Speech Compression.**

I.INTRODUCTION

There has been an unprecedented increase in the amount of digital data transmitted via networks especially through the internet and mobile cellular networks, over the last decade. Data compression offers an attractive approach to reducing communication cost by using available bandwidth effectively. Digital data represent text, images, video, sound etc. With this trend expected to continue, it makes sense to pursue research on developing algorithms that can be most effectively use available network bandwidth by maximally compressing data. Many methods in conjunction with BWT is discussed to achieve this. It has been observed that a preprocessing of the text prior to conventional compression will improve the compression efficiency much better. This paper presents a literature review which concentrates on lossless compression techniques for various types of data.BWT based methods along with entropy coding were considered in [1][2][3][4][5][6] performs significantly better than methods containing only entropy coding [7].The goal of this paper is to analyze the lossless compression techniques based on various types of data like audio, text files etc in terms of compression ratio, compression time, bit rate etc. And compare their results to ensure which technique gives the improved compression ratio

Manuscript received May 15, 2015.

Pratibha Warkade, M.E. Scholar, Dept. of Electronics and Telecommunication Engineering, Jabalpur Engineering College, Jabalpur, Madhya Pradesh, India

Agya Mishra, Assistant Professor, Dept. of Electronics and Telecommunication Engineering, Jabalpur Engineering College, Jabalpur, Madhya Pradesh, India

and rate of compression. The rest of the paper is organized as follows: Section II discusses various lossless speech compression techniques/algorithms used; Section III discusses about performance analysis; Section IV discusses about the comparison of existing algorithm; Section V concludes the paper and proposes the future work.

II. LOSSLESS SPEECH COMPRESSION TECHNIQUES

A. In this section we have discussed about the various lossless speech compression techniques/ algorithms used:-

Four [1] commonly used lossless compression algorithms are as follows: BWT, LZ-77, LZW, PPM.

Out of the four techniques used BWT+PPM gives best compression ratio for text and web and LZ77 provide less compression ratio for text ,good C.R for web compression .Hence LZ77 is quick, requires less power and memory.

B. Various **[2**] techniques are used to compress the speech signal/data as given below: BWT, MTF, Huffman, RLE, IDBE.

(i)*BWT Compression Algorithm:*It is a process that takes a block data and reorders it using a sorting algorithm.Its compression process look like this:

BWT< input-file/ MTF/RLE/ARI>output-file

The decompression process is just the reversible process and look like this:
UNARIinputfile/UNRLE/UNMTF/UNBWT>output-file.

(ii)*Intelligent Dictionary Based Coding(IDBE):* This type of coding provides more security to the data.

(iii)*MTF:* The main idea is to move to front the symbols that mostly occur, so those symbols will have smaller output number.

(iv)*Huffman Coding:* To avoid memory wastage and fast symbol searching; the length restricted Huffman coding can be used.

(v)*Context based speech coder*: The main objective of this speech coder is to develop a better transformation, yielding greater compression and added security through context based compression. Since the BWT has been preferred for this proposed coder as shown in fig.1.[2] has proved to be the most efficient method.

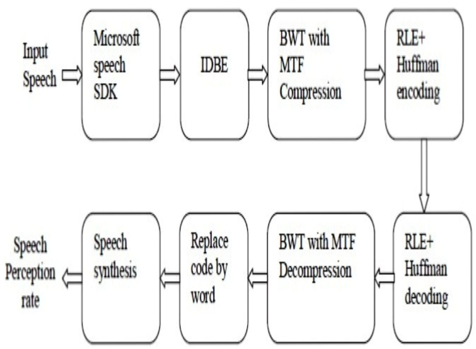

Fig 1: Context Based Speech Coder [2]

Out of the above mentioned techniques optimal one is the combination of BWT+IDBE+MTF+entropy coding as shown in the above Fig.1.[2].

C.Two techniques **[3]** are used BWT & MTF [2] as mentioned in the above paper; Scheme 2: applying lossless coding with MTF coding has lower bit rate and hence better compression ratio.

D. A simple lossless **[4]** audio data compression based on BWT has been done using BWT[2] & Huffman coding[2] or Arithmetic coding as explained below:

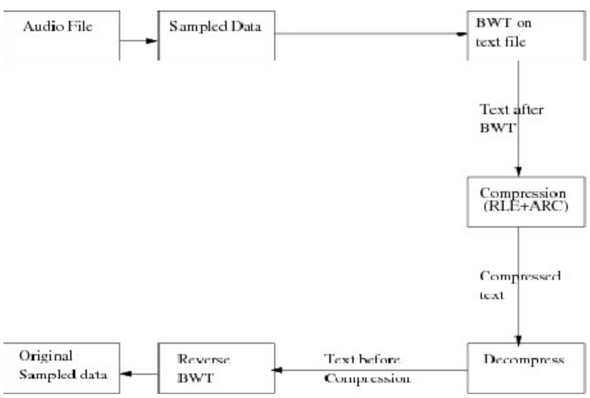

Fig 2: The Model For The Implementation Of BWT On Different Compression Algorithms For Lossless Data Compression [4]

From Fig.2 [4]. We process the text file with BWT and then on the transformed text we can apply any compression algorithm like Huffman/arithmetic coding to compress the data. We can store this compressed data along with the sample rate and the bits per sample as a representation of the audio file. In the reverse process we will use the operations in the reverse order to get back our original sampled data and then run the audio file with no degradation of sound quality. This process BWT+Huffman/Arithematic coding works for both floating and integer type data also.Hence overcome the drawback of [3].

E. A new algorithm **[5]** called j-bit encoding is used in combination with arithmetic coding & MTF that gives a best compression ratio.

F. In this paper **[6]** a new data structure method called FM-index has been described. This paper mainly tells about compression & indexing based on BWT & relationship between them.FM-index is a combination of BWT algorithm with the suffix array data structure and some auxiliary information. So FM-index has a great relationship with compression techniques based on BWT. Fig.3.[6] depicts the relationship between compression techniques and FM-index based on BWT.

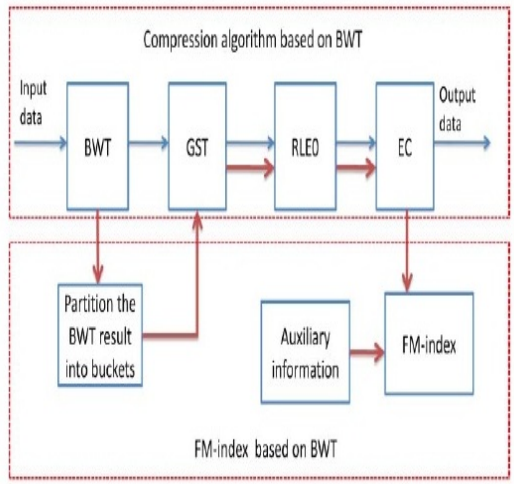

Fig 3: FM-Index And Compression Techniques Based On BWT[6]

Any method comprising BWT+F.M. provides less computational time.

G. Performance comparison **[7]** of Huffman & Lempel-ziv Welch data compression method is being done & explained as follows:-

The Huffman algorithm is able to reduce the data size by 43% on average, which is four times faster than the LZW algorithm.Hence, gives better result than LZW.

The BWT algorithm outperform the other classical algorithms by its effective compression capability and fastness. Especially for large files of text compression, the combination of BWT with other classical algorithms and word modeling gives an effective compression within reasonable resource cost.

III.PERFORMANCE ANALYSIS OF DIFFERENT TECHNIQUES

In this section a performance analysis of lossless speech compression techniques presented in different reference papers is done.

In **[1]** four graphs are plotted to test compression ratio, static memory & time for text and web both as shown below:

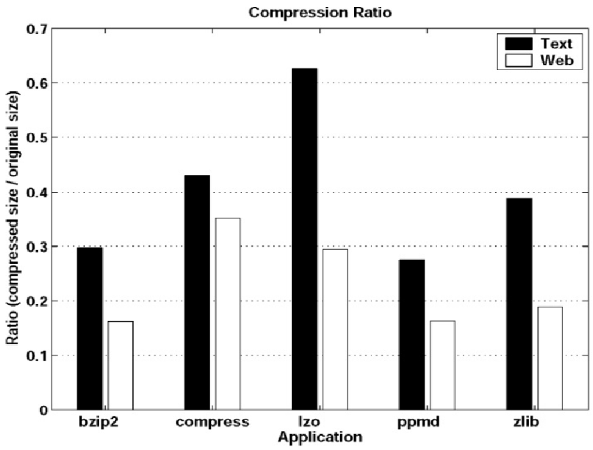

Fig 4: Results From Compression Ratio Tests [1]

Fig.4.[1] authors concluded that bzip2+ppmd gives best compression ratio i.e. 0.29 rather than lzo (0.64).

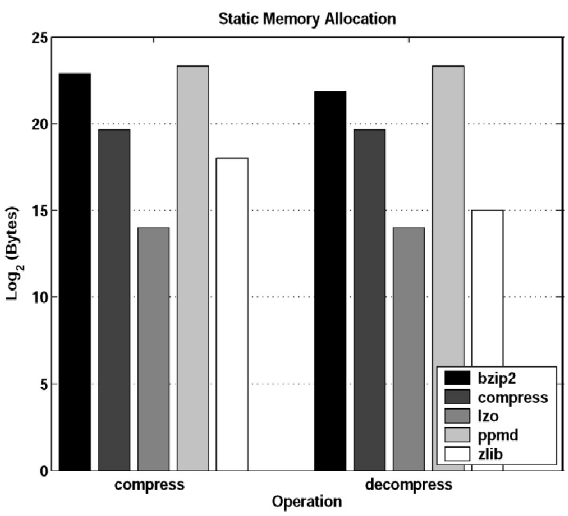

Fig 5: Results From Memory Tests [1]

Fig.5.[1] authors concluded that LZO requires least static memory.

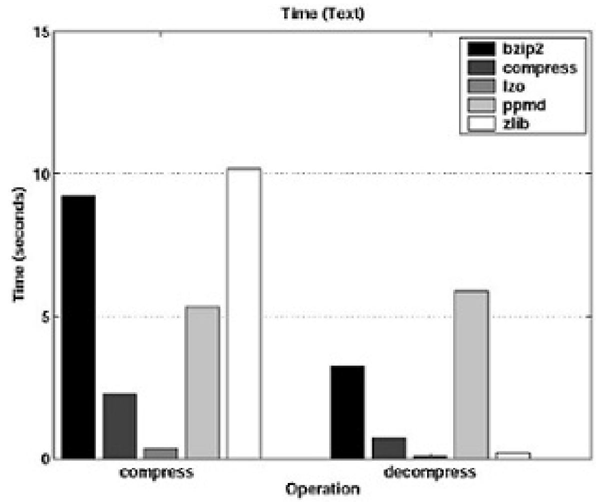

Fig 6: Results from Completion Time (Text) Tests [1]

Fig.6.[1] authors concluded that LZO requires less computational time for text.

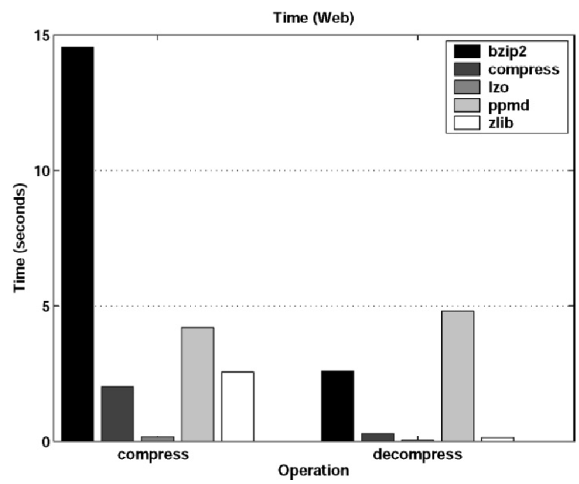

Fig 7: Results From Completion Time (Web) Tests[1]

Fig.7[1] authors concluded that LZO requires less computational time hence can be implemented in any mobile devices.
In **[2]** Bit rate& compression ratio are the parameter measured as shown below in the following graph:-

1.*Bit Rate*:Fig.8.[2]shows the bit rate of the proposed coder. The proposed coder achieves an average of 37bps bit rate which is less than the average bit rate of IDBE and Huffman coding alone.

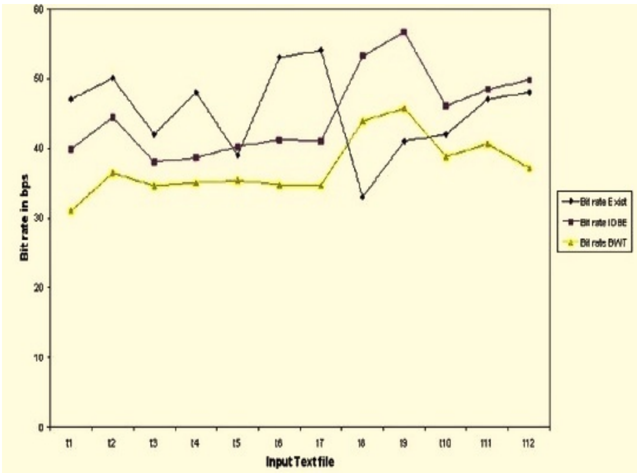

Fig 8:Graph Between Input Text File And Bit Rate In bps [2]

2. *Compression Ratio*:

Fig.9.[2] shows the compression ratio of the proposed coder with existing Huffman coding alone, IDBE alone and with the combination of BWT, IDBE and Huffman coding as 3.10, 2.77 and 3.31respectively.

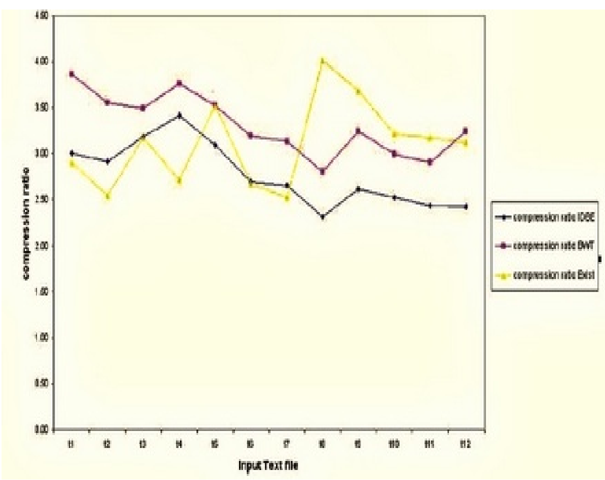

Fig 9: Graph Between Input Text File And Compression Ratio [2]

The proposed coder achieves an average of 37bps bit rate with the compression ratio of 3.31:1, which will be improved when this coder is used for very large size of text files.

In **[3]** Bit rate is the parameter measured in the paper as shown below in the following graph:-

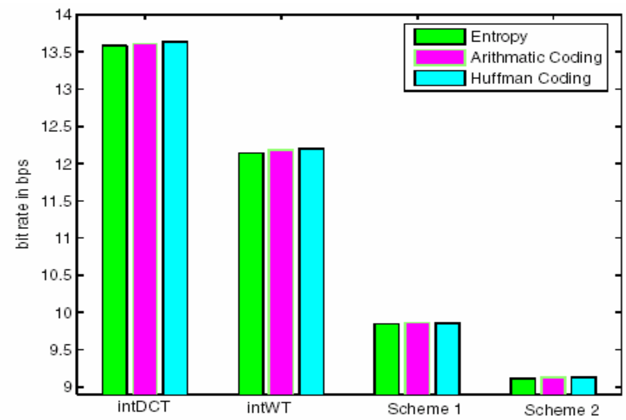

Fig.10.Bit Rates For Various Lossless Coding Schemes, Fs=44.1 KHz[3]

In this paper, authors proposed a new lossless audio coding scheme using BWT & MTF . Fig.10[3] shows that combined scheme 2achieved a great improvement in bit rate(9.1bps) over other techniques used.

In **[4]** authors proposed a shorten technique of lossless audio compression with BWT based method for 8 files as shown below:-

Table 1: Comparison of 8 different audio files with different size, sample rate, bits and compression ratio [4]

File	\Original Size	Shorten		BWT Based	
		Compressed size	CR	Compressed size	CR
File1	183	115	37.16	112	38.8
File2	246	149	39.43	143	41.87
File3	176	108	38.64	110	37.5
File4	208	136	34.62	132	36.54
File5	351	220	37.32	208	40.74
File6	184	112	39.13	114	38.04
File7	283	169	40.28	158	44.17
File8	324	216	33.33	190	41.36
Avg	244.36	153.13	37.33	145.88	40.3

From Table 1[4] BWT based method has shown better C.R. with almost all audio files we have considered for experiment. This method works fine with the audio containing floating point values too, which removes the drawbacks of previous method on this technique[3].

In **[5]** authors proposed five combinations of data compression algorithm that are used to find out which combination gives the best compression ratio:

That is RLE+BWT+MTF+**JBE**+ARI.

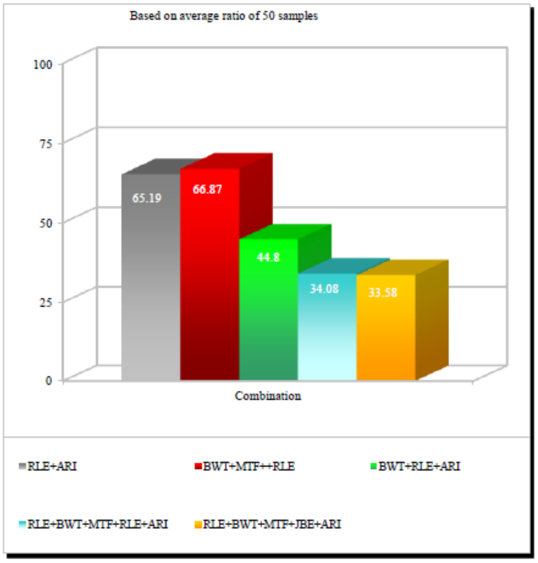

Fig 11: Average Ratio Of Text Files [5]

Fig.11 [5] shows that text files are compressed with better compression ratio (33.58%) by algorithms that combined with J-bit encoding.

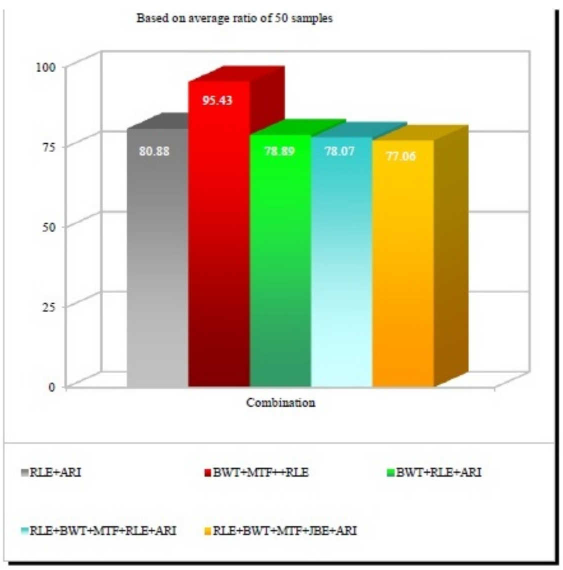

Fig 12: Average Ratio Of Audio Files[5]

Fig.12.[5]shows that wave audio files are compressed with better compression ratio (77.06%) by algorithms that combined with J-bit encoding.

In **[6];** authors has compared several tools which are widely used including gzip (v1.2.4), szip (v1.12a)[20],bzip2(v 1.0.6)[1],bicom(v 1.01). Bzip2 and szip are based on BWT, gzip is based on LZ77 and bicom is based on PPM. The test data is Large Corpus and the results are shown in table 2.

Table 2:Compression rates (bits/byte) for large corpus[6]

File	File size	Bicom	Szip	Bzip2	gzip
Large.txt	4,047,392	1.69	**1.63**	1.67	2.35
E.Coli	4,638,690	2.12	**2.02**	2.16	2.31
World192.txt	2,473,400	**1.44**	1.60	1.58	2.34

From Table 2[6], compression tools based on BWT perform much better than tools based on LZ77 for all files, and even better than bicom based on PPM for the first two files. In general, tools based on PPM have a better compression rate with much higher time consumption than tools based on BWT.

Table 3:Varying read length using Bowtie, Maq and SOAP [6]

Read length	Program	CPU time	Peak memory (megabytes)	Speed-up	Reads aligned
36 bp	Bowtie	6m15s	1,305	-62.2	
	Maq	3h52m26s	804	36.7x	65.0
	Bowtie –v 2	4m55s	1,138	-	55.0
	SOAP	16h44m3s	13,619	216x	55.1
50 bp	Bowtie	7m11s	1,310	-	67.5
	Maq	2h39m56s	804	21.8x	67.9
	Bowtie –v 2	5m32s	1,138	-	56.2
	SOAP	48h42m4s	13,619	691x	56.2
76 bp	Bowtie	18m58s	1,323	-	44.5
	Maq 0.7.1	4h45m7s	1,155	14.9x	44.9
	Bowtie –v 2	7m35s	1,138	-	31.7
	SOAP	do not support			

From table 3[6], authors concluded that Bowtie based on FM-index has a great advantage over traditional tools (Maq, SOAP) in time consumption. Hence methods based on BWT+F.M. provides better result.

In [7]; we have considered 2 tables for comparing the compression performance & time for 4 different compression algorithms:-

Table4: Compression performance of Huffman, LZW, HLZ and LZH techniques [7]

Data type	Size before compression (Bits)	Size after compression (Bits)				Compression ratio				Saving (%)			
		Huffman	LZW	HLZ	LZH	Huffman	LZW	HLZ	LZH	Huffman	LZW	HLZ	LZH
Temperature	200	106	200	296	106	0.53	1.00	1.48	0.53	47.00	0.00	-48.00	47.00
	400	247	400	544	247	0.62	1.00	1.36	0.62	38.25	0.00	-36.00	38.25
	600	398	592	776	396	0.66	0.99	1.30	0.66	34.00	1.33	-29.33	34.00
	800	550	784	992	546	0.69	0.98	1.24	0.68	31.25	2.00	-24.00	31.75
Humidity	200	102	200	272	102	0.51	1.00	1.36	0.51	49.00	0.00	-36.00	49.00
	400	240	400	536	240	0.60	1.00	1.34	0.60	40.00	0.00	-34.00	40.00
	600	363	584	720	363	0.61	0.97	1.20	0.61	39.50	2.67	-20.17	39.50
	800	485	752	896	488	0.61	0.94	1.12	0.61	39.38	6.00	-12.00	39.00
ECG	200	92	184	264	88	0.46	0.92	1.32	0.44	54.00	8.00	-32.00	56.00
	400	243	384	536	237	0.61	0.96	1.34	0.59	39.25	4.00	-34.00	40.75
	600	411	584	800	404	0.69	0.97	1.33	0.67	31.50	2.67	-33.33	32.67
	800	555	776	1000	549	0.69	0.97	1.25	0.69	30.63	3.00	-25.00	31.38
Text	800	367	504	728	328	0.46	0.63	0.91	0.41	54.13	37.00	9.00	59.00
	1200	567	696	1000	491	0.47	0.58	0.83	0.41	52.75	42.00	16.67	59.08
	1600	753	840	1264	626	0.47	0.53	0.79	0.39	52.94	47.50	21.00	60.88
	2000	936	960	1480	743	0.47	0.48	0.74	0.37	53.20	52.00	26.00	62.85
	Average									42.92	13.01	-18.20	45.07

From **Table 4[7]** authors concluded that: LZW performs well for text data sizes of 800 bits, with a saving percentage of 37% being observed. For double compression, the LZH performs better as compared to the HLZ. However, the LZH algorithm gives better compression since the output from LZW contains highly repetitive value. This repeated value is suitable for Huffman compressions.

Table5: Compression time of Huffman, LZW, HLZ and LZH [7]

Data type	Size before Compression (Bits)	Time taken (sec) Huffman		LZW		HLZ		LZH	
		Encoder	Decoder	Encoder	Decoder	Encoder	Decoder	Encoder	Decoder
Temperature	200	0.143	0.073	0.360	0.027	0.733	0.143	0.492	0.053
	400	0.790	0.183	0.848	0.119	2.166	0.916	1.040	0.209
	600	0.481	0.669	1.298	0.098	3.568	4.587	1.684	0.353
	800	0.313	1.225	2.102	0.120	3.950	6.957	2.009	0.445
Humidity	200	0.207	0.065	0.509	0.029	0.790	0.163	0.543	0.059
	400	0.341	0.569	1.506	0.059	4.098	0.518	0.783	0.231
	600	0.230	0.279	1.863	0.096	4.036	3.838	1.473	0.229
	800	0.648	0.505	1.805	0.292	2.923	6.339	3.181	0.558
ECG	200	0.187	0.072	0.748	0.043	1.237	0.163	1.156	0.058
	400	0.586	0.300	1.151	0.068	3.814	0.531	1.429	0.137
	600	0.650	0.403	3.284	0.084	4.923	4.317	2.362	0.311
	800	0.506	0.943	2.581	0.191	3.132	7.605	4.171	0.582
Text	200	0.178	0.053	0.697	0.055	0.702	0.147	0.823	0.054
	400	0.222	0.135	1.730	0.075	3.294	0.341	1.462	0.098
	600	0.447	0.106	1.984	0.107	4.316	0.629	2.424	0.175
	800	0.446	0.136	2.046	0.171	3.837	3.263	1.926	0.372
	Average	0.398	0.357	1.532	0.102	2.970	2.529	1.685	0.245

Table 5 [7]shows the result of time taken to compress and decompress different data using all four algorithms.The Huffman algorithm only takes 0.398 sec, while LZW algorithm takes 1.532 sec.

IV. Table 6: COMPARISON OF DIFFERENT LOSSLESS SPEECH COMPRESSION TECHNIQUES

S.No.	Ref. Paper	Algorithm	Measuring parameter	Advantages	Limitations
1	Evaluating lossless data compression algorithm for use on mobile devices[1]	Lempelziv-77+Lempal ziv-wheeler+prediction with partial match ,Burrows wheeler transform (optimal LZ-77)	Compression ratio: -lzo =0.62 for text files -zlib =0.38 for text files	quick process requires low processing power & low memory	Hardware limited (specific)
2	A word level context based speech scheme [2]	BWT+ MTF/RLE +Huffman encoding	-C.R =3.31:1, bitrate =37bps, recognigation rate, perception rate.	Less bandwidth than compressed voice, less communication cost with acceptable perceptual quality & security.	Gives less C.R for smaller size of text files
3	Lossless audio coding using BWT & MTF[3]	BWT .MTF & entropy coder: -scheme 1: BWT without MTF -scheme 2:BWT with MTF	For scheme 2)BWT+MTF: -bitrate =9bps for Fs=44.1KHz;	Scheme2: BWT with MTF has lower bit rate & better compression ratio.	-Limited to integer data only -not applicable for floating point values

4	A simple lossless audio datacompressio nbased on BWT[4]	BWT,Huffman , arithematic coding.	-C.R values 40.3 bps (avg. value) for all files from file 1 to file 12.	-better audio quality -used for compression of text files -it works for both integer & floating point values too.	This method is applicable for only waveform audio files
5	A new algorithm for data compression optimization[5]	JBE (j-bit encoding)	C.R for text files= 33.55% Audio=77.06%	Gives best C.R for combination of JBE+MTF+ARI+BWT+RLE	Depends on specific content of audio files
6	Compression & indexing based on BWT :A survey[6]	-BWT+ F.M index : (F.M index combination of BWT + suffix array) -compression tools based on BWT/LZ77/ PPM	Compression rate of file : (bits/bytes) a)large text =1.63 bits b)E. coli=2.02 bits	- F.M index :do not depend upon alphabet size -it scales well with size of the alphabet - compression tools :szip & bzip based on BWT has better rate & take less computational time .	F.M index depends upon the alphabet size for a limited time –bounds.
7	Performance comparison of Huffman & LZW datacompre -ssion for wireless sensor node application [7]	Huffman coding ,LZW, HLZ, LZH	Average Computational time : Huffman codes takes 0.398 Sec, LZW takes 1.532 sec. From table 4: LZW has compression ratio of 0.41 bits. From table 5:LZW has 0.102 sec of computational time & Huffman has 0.357 sec.	From Table 4:- -Huffman is better than LZW; - LZW performs well for text data sizes of 800 bits, with a saving percentage of 37%. -For double compression:- LZH algorithm gives better compression since the output from LZW contains a highly repetitive values. From Table 5:- - The Huffman algorithm only takes 0.398 sec, while LZW algorithm takes 1.532 sec. This is due to the Huffman algorithm being less complex than the LZW algorithm,which means it takes less time to compress the data. - For the decompression part, the average time taken for the LZW is less than for the Huffman .	-LZW perform bit by bit scanning which results in an increase in output bits. -this process takes long time for double compression.

BWT=Burrows Wheeler Transform LZW=Lempel-Ziv Welch PPM=Prediction With Partial
MTF=Move to Front Transform LZ-77=Lempel-Ziv 77 ARI=Arithematic coding
RLE=Run Length Encoding JBE=j-bit encoding LZH=Lempel-Ziv followed by Huffman
C.R=Compression Ratio HLZ=Huffman followed by LZW bps=bits per second
Fs=Sampling frequency

Above table concludes that LZ77[1] is the optimal technique used in mobile system since it requires less time & less memory but depends upon hardware specifications.BWT+ MTF/RLE +Huffman [2]encoding gives best C.R for smaller size of text files. Scheme 2: BWT with MTF [3] has lower bit rate & better C.R but Limited to integer data only. A simple lossless audio data compression based on BWT [4]provides better audio quality used for compression of text files it works for both integer & floating point values too. JBE (j-bit encoding) [5] gives best C.R for combination of JBE+MTF+ARI but depends on specific contents of audio files.BWT+ F.M index gives[6]Compression rate of file: (bits/bytes)large text =1.63 bits & E. coli=2.02 bits. Compression tools:- szip & bzip based on BWT has better rate & take less computational time, but F.M. index depends upon the alphabet size for a limited time interval. This study analyses the performance of the Huffman and Lempel-Ziv Welch(LZW) algorithms [7] when compressing data that are commonly used in Wireless Sensor Network. From the experimental results, the Huffman algorithm gives a better performance when compared to the LZW algorithm for this type of data size by 43% on average, which is four times faster than the LZW algorithm. But it still has the problem observed in above table, so it is necessary to use any entropy coding method along with BWT to get remedy of this problem.

V. CONCLUSION

In this paper, a widespread literature survey on various lossless speech compression techniques for different data files has been done. We have presented a comparative analysis of these techniques for various data files i.e. text and audio both. Among all these techniques proposed for compressing data, it is rather difficult to determine a single method that clearly outperforms the rest.BWT+MTF+Huffman encoding[2] gives best C.R but limited to smaller size of text files.BWT+MTF[3] based method is limited to integer data only.In[4] it overcomes the drawback of [3] and works for both integer and floating points value too in order to get better C.R.BWT+MTF+JBE+ARI based method are used to compress the audio files but limited to specific contents of it.BWT+F.M. based techniques works for both smaller and larger text files hence overcomes the drawback of [2].In[7] Huffman algorithm is four times better than LZW since it reduces data size by 43% on average.BWT is a very good lossless compression technique which can be integrated with any entropy coder in order to get improved compression ratio for better result in future work.

REFERENCES

[1] Paul Brittan (BRTPAU008),"Evaluating Lossless Data Compression Algorithms for Use on Mobile Devices" , Literature Synthesis ,2008.

[2] Chapter6:A Word Level Context Based Speech Coding Scheme.

[3] Hend Ali Elsayed and Masoud Alghoniemy, "Lossless Audio Coding using Burrows-Wheeler Transform and Move-to-Front Coding", Department of Electrical Engineering ,University of Alexandria, Alexandria, 21544 Egypt,IEEE 2006.

[4] Mrinal Kumar Borah, V.K. Govindan, "A Simple Lossless Audio Data Compression Based On Burrows Wheeler Transform",Department of Computer Science & Engineering, National Institute of Technology, Calicut, Calicut, India,April-2013IEEE.

[5] I Made Agus Dwi Suarjaya "A New Algorithm for Data Compression Optimization" , Information Technology Department Udayana University Bali, Indonesia,2012.

[6] Dengfeng Zhang, Qing Liu+,Yuewan Wu, Yaping Li, Lin Xiao,"Compression and Indexing Based on BWT: A Survey", School of Information, Renmin University of China,2013IEEE..

[7] Asral Bahari Jambek and Nor Alina Khairi "Performance Comparison Of Huffman And Lempel-Ziv Welch Data Compression For Wireless Sensor Node Application",School of Microelectronic Engineering, Universiti Malaysia Perlis, Pauh Putra Campus, Perlis, Malaysia,2014.

[8] M.P.Bhuyan, V.Deka, S.Bordoloi "Burrows Wheeler Based Data Compression And Secure Transmission",Department of Information Technology, Gauhati University. Department of Information Technology, Gauhati University.Department of Computer Applications, Assam Engineering College,Dec-2013.

[9] Rupinder Singh Brar & Bikramjeet singh "A Survey on Different Compression Techniques and Bit Reduction Algorithm for Compression of Text/Lossless Data", International Journal of AdvancedResearch in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013.

[10] S.R. kodituwakku & U. S.Amarasinghe, **"**Comparison Of Lossless Data Compression Algorithms For Text Data", Indian Journal of Computer Science and Engineering,Vol 1 No 4 416-425.

Pratibha Warkade, M.E. Scholar, Dept. of Electronics & Telecommunication Engineering ,Jabalpur Engineering College, Jabalpur, India. And has done Bachelor of Engineering from Jabalpur Engineering College, Jabalpur, India

Dr.Agya Mishra working as Assistant Professor in Dept. of Electronics & Telecommunication Engineering,, Jabalpur Engineering College, Jabalpur, India. She has done her Ph.D. and M.Tech from Maulana Azad National Institute of Technology, Bhopal, India. And Bachelor of Engineering from Government Engineering College, Ujjain, India. Her specialization are in the field of Digital Signal & Image Processing, Higher Order Neural Network, Wavelet transform applications, Computational Intelligence etc.